

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій
Кафедра систем штучного інтелекту



Лабораторна робота №6
з курсу “Дискретна математика ”

Виконав:
ст. гр. КН-110
Петровський Олександр

Викладач:
Мельникова Н.І.

Львів – 2018

Тема:**«Генерація комбінаторних конфігурацій»****Мета роботи:**

Набуття практичних вмінь та навичок при комп'ютерній реалізації комбінаторних задач.

Теоретичні відомості:

Головна задача комбінаторики – підрахунок та перелік елементів у скінчених множинах.

Правило додавання: якщо елемент – x може бути вибрано n способами, а у іншими m способами, тоді

вибір „ x або y “ може бути здійснено $(n+m)$ способами.

Правило добутку: якщо елемент – x може бути вибрано n способами, після чого y – m способами, тоді

вибір упорядкованої пари (x, y) може бути здійснено $(n \cdot m)$ способами.

Набір елементів x_1, x_2, \dots, x_m з множини $X = \{x_1, x_2, \dots, x_n\}$ називається вибіркою об'єму m з n елементів – (n, m) – вибіркою.

Упорядкована (n, m) – вибірка, в якій елементи не можуть повторюватися, називається (n, m) – розміщенням A_n^m .

Упорядкована (n, m) – вибірка, в якій елементи можуть повторюватися, називається (n, m) – розміщенням з повторюваннями \bar{A}_n^m .

Неупорядкована (n, m) – вибірка, в якій елементи не можуть повторюватися, називається (n, m) – сполученням A_n^m .

Неупорядкована (n, m) – вибірка, в якій елементи можуть повторюватися, називається (n, m) – сполученням з повторюваннями.

A_n^n – називається перестановкою \bar{C}_n^m .

Лексикографічний порядок – це природний спосіб упорядкування послідовностей на основі порівняння індивідуальних символів. На множині всіх розміщень із r елементів означимо порівняння таким чином:

$b_1 b_2 \dots b_r < a_1 a_2 \dots a_r$, якщо $\exists m : (b_i = a_i, i < m) \wedge (b_m < a_m)$.

Варіант № 5

Завдання 1:

- Скільки різних кілець, що світяться, можна утворити, розмістивши по колу 10 ріznокольорових лампочок (кільця вважати однаковими, якщо послідовність кольорів одна й та сама)?

При створенні кільця з 10 лампочок ми можемо поставити 1 з 10 на перше місце, 1 з 9 на друге і т. д. Отже, ми маємо $10!$ (або P_{10}). Але задля того, щоб уникнути повторів, пов'язаних з циклічністю, поділимо відповідь на 9.

$$P_{10}/9 = 3628800/9 = 403200$$

- На дев'яти картинках записані цифри 1, 2, 3, 4, 5, 6, 7, 8, 9 (на кожній картці по одній цифрі). Беруть чотири катки і складають з них чотирицифрове число. Скільки різних чисел можна отримати таким чином?

У цьому випадку маємо розміщення без повторень з 9 по 4 елементи.

$$A_9^4 = 3024$$

- Скільки існує трикутників, довжини сторін яких мають одне з таких значень: 4, 5, 6, 7 см?

У цьому випадку маємо сполучення з повтореннями з 4 по 3 елементи.

$$\bar{C}_4^3 = C_{4+3-1}^3 = 20$$

- Скільки різних правильних нескоротних дробів можна скласти з чисел 2, 5, 7, 11, 15, 17, 19, 23, 25 так, щоб у кожен дріб входило два числа?

Оскільки дріб складається з двох елементів множини кількість правильних нескоротніх дробів буде дорівнювати $8!$ А кількість правильних нескоротних:

$$8*5(7 - 2 \text{ скоротних})*6*5*3(4 - 1 \text{ скоротний})*3*2 = 21600$$

- Скільки п'ятицифрових чисел можна утворити з цифр 2, 3, 6, 7, 8 (без повторення) так, щоб парні цифри не стояли поруч?

У нас є 6 варіантів розміщення парних і 2 – непарних чисел. Тобто:

$$A_3^2 * A_2^2 = 6 * 2 = 12$$

- Скількома способами можна розкласти 28 різних предметів у чотири однакові ящики так, щоб у кожному з них опинилося по 7 предметів?

Для розкладання ми спочатку маємо вибрати 7 з 28, потім 7 з 21 і 7 з 14.

$$C_{28}^7 * C_{21}^7 * C_{14}^7 = 1184040 * 116280 * 3432 = 472518347558400$$

7. Знайти кількість цілих додатних чисел, що не більше 1000 і не діляться на одне з чисел 6, 7 і 15.

Кількість чисел, які кратні 6 – 166, з них 23 – 7, 33 – 15, 4 – 7 і 15

Кількість чисел, які кратні 7 – 142, з них 23 – 6, 9 – 15, 4 – 6 і 15

Кількість чисел, які кратні 15 – 66, з них 33 – 6, 9 – 7, 4 – 6 і 7

Тому за формулою включен-виключень знайдемо:

$$166 + 142 + 66 - (23 + 33 + 9) + 4 = 313$$

Програма:

Задані додатні цілі числа n та r. Побудувати у лексикографічному порядку всі розміщення з повтореннями із r елементів множини {1, 2, ..., n}. Побудувати розклад $(x + y)^r$.

Код:

```
#include <cs50.h>
#include <stdio.h>
#include <stdlib.h>

//VARS///////////////////////////////
///////////////////////////////

short set[10];
short set_len;

//PROTOTYPES///////////////////////
///////////////////////////////

void display(short n, short r);
void print_set();
void draw_a(short n, short r);
void permutation();
void binominal(short n);
short fact (short in);
short comb(short n, short k);
void bin(short n);
```

```
//MAIN//////////MAIN//
```

```
int main(){
```

```
    permutation();
```

```
    binomial(7);
```

```
}
```

```
//FUNCTIONS//////////FUNCTIONS//
```

```
//PERMUTATION PART
```

```
void display(short n, short r){
```

```
    print_set();
```

```
    set[set_len-1]++;
```

```
    for (short i = set_len; i > 0; i--){
```

```
        if (set[i] > n){
```

```
            set[i] = 1;
```

```
            set[i-1]++;
```

```
        }
```

```
}
```

```
        if (set[0] != n+1) display(n, r);
```

}

```
void print_set(){

    printf(" { ");

    for (short i = 0; i < set_len; i++) printf("%i ", set[i]);

    printf("} ");

}
```

```
void draw_a(short n, short r){

    printf("\n");
    printf(" ~~\n");
    printf(" /\%i____\n", r);
    printf(" /__\\____\n");
    printf("/ \\%i\n", n);
    printf("\n");

}
```

```
void permutation(){

    printf("\n\x1b[32mPERMUTATION PART\x1b[0m\n\n");

    printf("N value is ");
    short n = GetInt();
    printf("N value is ");
    short r = GetInt();

    set_len = r;
```

```
for (short i = 0; i < set_len; i++) set[i] = 1;

draw_a(n, r);
display(n, r);
printf("\n");

}
```

//BINOMINAL PART

```
void binominal(short n){

printf("\n\x1b[32mBINOMINAL PART\x1b[0m\n\n");

printf("x^%i", n);

for (int i = 1; i < n; i++){

printf(" + %i(x)^%i(y)^%i", comb(n, i), n-i, i);

}

printf(" + y^%i\n\n", n);

}
```

```
short fact(short in){
```

```
    short out = 1;
    for(short i = in; i>1; i--){
        out*=i;
    }
```

```
    return out;  
}  
  
short comb(short n, short k){  
  
    return fact(n)/(fact(k)*fact(n-k));  
  
}
```

Висновки:

Я набув практичних вмінь та навичок при комп'ютерній реалізації комбінаторних задач.